UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/595,348 | 04/11/2006 | Groschel Andreas | 72165 | 7989 |

23872 7590 08/17/2010
MCGLEW & TUTTLE, PC
P.O. BOX 9227
SCARBOROUGH STATION
SCARBOROUGH, NY 10510-9227

| EXAMINER | |
|---|---|
| ZHEN, LI B | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2194 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 08/17/2010 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| *Office Action Summary* | **Application No.** 10/595,348 | **Applicant(s)** ANDREAS ET AL. | |
|---|---|---|---|
| | **Examiner** LI B. ZHEN | **Art Unit** 2194 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on _03 June 2010_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) _1-5,7-23 and 25-32_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-5,7-23 and 25-32_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

## DETAILED ACTION

1. This office action is in response to the amendments filed 6/3/2010. Claims 1 – 5, 7 – 23, and 25 – 32 are pending in the application.

### *Response to Amendment*

2. Amendment to the claims overcome the previous claim rejections under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. **Claims 1 – 4, 7 – 10, 12, 13, 15, 17 - 19, 21, 22, 31 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,996,828 to Kimura et al. [hereinafter Kimura] in view of US 20010016879 A1 to Sekiguchi et al. [hereinafter Sekiguchi].**

5. As to claim 1, Kimura teaches a method for operating (implementing) a secondary operating system [second OS; col. 5, lines 5 – 9] on a processor [col. 3, line

64 – col. 4, line 3 and col. 4, lines 20 – 22] in addition to a primary operating system [first OS; col. 4, lines 60 – 65], the method comprising the steps of:

switching from the primary operating system to the secondary operating system based on an interrupt call [switching of the running OS and starting of the interrupt processing of the second OS; col. 16, lines 52 – 65; col. 13, lines 46 – 56; col. 15, line 58 – col. 16, line 5];

loading a secondary operating system driver (SOS driver) of the primary operating system [control for running the multi-OS is incorporated as a device driver for the first OS (hereinafter referred to as multi-OS driver); col. 10, lines 19 – 22 and 54 – 60; col. 11, lines 14 – 20; col. 12, lines 4 – 10] and activating said secondary operating system driver for loading [loading the second OS by execution of the multi-OS management program in the first OS, Fig. 15; col. 12, lines 24 – col. 13, line 13] and controlling said secondary operating system [col. 12, lines 11 – 23 and 45 – 52; col. 14, lines 5 – 20; col. 15, lines 13 – 36], said secondary operating system driver comprising an interrupt handling routine [interrupt table 107, the interrupt management program, the interrupt handler, the interface module to be referred from each operating system and the like are stored in the common area as part of the multi-OS driver; col. 11, lines 21 – 32]; and

determining information [address of the interrupt handler of the second OS; col. 16, lines 42 – 51; col. 6, lines 17 – 27] stored in an interrupt table of the secondary operating system with said interrupt handling routine, said information corresponding to a point in the secondary operating system at which said interrupt call is to be serviced

[interrupt handler in the common area 203 captures the interrupt and decides an operating system to process the interrupt with reference to the processing OS 1621 of the interrupt identification table 1620. Then, the control is passed to the address of the handler 1622; col. 13, lines 46 – 56; col. 13, lines 24 – 25; col. 13, lines 45 – 63]. Kimura teaches changing the interrupt table register value to the address of the interrupt table assigned to the multi-OS driver [col. 13, lines 5 – 13 and col. 4, lines 20 – 23]. Kimura does not specifically teach exchanging interrupt tables of the operating systems during said interrupt call.

However, Sekiguchi teaches exchanging interrupt tables [portions essential also for the second OS to access: such as the interrupt table 107; paragraphs 0218 and 0239] of the operating systems [interrupt table in the first OS main memory area is changed to the interrupt table 107 of the support driver; paragraphs 0152 and 0146] during said interrupt call [an external interrupt occurs; paragraph 0158].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the invention of Kimura to incorporate the features of Sekiguchi. One of ordinary skill in the art would have been motivated to make the combination because this provides concurrent execution of a plurality of OSs without using specific hardware, by modifying an initializing process and interrupt management of each OS and by adding an interrupt management program [paragraph 0016 of Sekiguchi].

6. As to claim 22, Kimura as modified by Sekiguchi teaches a device for operating a secondary operating system [second OS; col. 5, lines 5 – 9 of Kimura] on a processor [col. 3, line 64 – col. 4, line 3 and col. 4, lines 20 – 22 of Kimura] in addition to a primary operating system [first OS; col. 4, lines 60 – 65 of Kimura], the device comprising:

a means for switching from the primary operating system to the secondary operating system based on an interrupt call [switching of the running OS and starting of the interrupt processing of the second OS; col. 16, lines 52 – 65; col. 13, lines 46 – 56; col. 15, line 58 – col. 16, line 5 of Kimura];

a secondary operating system driver (SOS driver) of the primary operating system [control for running the multi-OS is incorporated as a device driver for the first OS (hereinafter referred to as multi-OS driver); col. 10, lines 19 – 22 and 54 – 60; col. 11, lines 14 – 20; col. 12, lines 4 – 10 of Kimura] for loading [loading the second OS by execution of the multi-OS management program in the first OS, Fig. 15; col. 12, lines 24 – col. 13, line 13 of Kimura] and controlling the secondary operating system is constructed [col. 12, lines 11 – 23 and 45 – 52; col. 14, lines 5 – 20; col. 15, lines 13 – 36 of Kimura];

a function for exchanging interrupt tables [portions essential also for the second OS to access: such as the interrupt table 107; paragraphs 0218 and 0239 of Sekiguchi] of the operating systems [interrupt table in the first OS main memory area is changed to the interrupt table 107 of the support driver; paragraphs 0152, 0146, and 0158 of Sekiguchi], said secondary operating system driver comprising an interrupt service routine [interrupt table 107, the interrupt management program, the interrupt handler,

the interface module to be referred from each operating system and the like are stored in the common area as part of the multi-OS driver; col. 11, lines 21 – 32 of Kimura], said interrupt service routine determining information [address of the interrupt handler of the second OS; col. 16, lines 42 – 51; col. 6, lines 17 – 27 of Kimura] stored in an interrupt table of the secondary operating system corresponding to a point in the secondary operating system at which said interrupt call is to be serviced [interrupt handler in the common area 203 captures the interrupt and decides an operating system to process the interrupt with reference to the processing OS 1621 of the interrupt identification table 1620. Then, the control is passed to the address of the handler 1622; col. 13, lines 46 – 56; col. 13, lines 24 – 25; col. 13, lines 45 – 63 of Kimura].

7. As to claim 31, Kimura as modified teaches a method for operating (implementing) a secondary operating system [second OS; col. 5, lines 5 – 9 of Kimura] on a processor [col. 3, line 64 – col. 4, line 3 and col. 4, lines 20 – 22 of Kimura] in addition to a primary operating system [first OS; col. 4, lines 60 – 65 of Kimura] in which a change from the primary operating system to the secondary operating system takes place through an interrupt call [col. 16, lines 52 – 65; col. 13, lines 46 – 56; col. 15, line 58 – col. 16, line 5 of Kimura], the method comprising the steps of:

providing an interrupt call [external interrupt occurs; col. 15, line 57 – col. 16, line 11 of Kimura];

switching from the primary operating system to the secondary system based on said interrupt call [switching of the running OS and starting of the interrupt processing of

the second OS; col. 16, lines 52 – 65; col. 13, lines 46 – 56; col. 15, line 58 – col. 16, line 5 of Kimura];

providing the primary operating system with a secondary operating system driver [control for running the multi-OS is incorporated as a device driver for the first OS (hereinafter referred to as multi-OS driver); col. 10, lines 19 – 22 and 54 – 60; col. 11, lines 14 – 20; col. 12, lines 4 – 10 of Kimura], said secondary operating system driver comprising an interrupt call servicing routine [interrupt table 107, the interrupt management program, the interrupt handler, the interface module to be referred from each operating system and the like are stored in the common area as part of the multi-OS driver; col. 11, lines 21 – 32 of Kimura];

loading said secondary operating system driver (SOS driver) in the primary operating system and activating said secondary operating system driver [first operating system loads the object file of the multi-OS driver; col. 11, lines 13 – 21 of Kimura] for loading and controlling the secondary operating system [col. 12, lines 24 – col. 13, line 13; col. 12, lines 11 – 23 and 45 – 52; col. 14, lines 5 – 20; col. 15, lines 13 – 36 of Kimura], said primary operating system having a primary operating system interrupt table [interrupt table in the first OS main memory area; paragraphs 0152, 0146, and 0158 of Sekiguchi], said secondary operating system having a secondary operating system interrupt table [portions essential also for the second OS to access: such as the interrupt table 107; paragraphs 0218 and 0239 of Sekiguchi];

replacing said primary operating interrupt table with said second operating system interrupt table during said interrupt call [interrupt table in the first OS main

memory area is changed to the interrupt table 107 of the support driver; paragraphs 0152, 0146, and 0158 of Sekiguchi];

processing information stored in said second operating system interrupt table with said secondary operating system driver [At Step 1811, the interrupt handler of the second OS is started; col. 16, lines 42 – 52 of Kimura] such that said interrupt call servicing routine determines the information [address of the interrupt handler of the second OS; col. 16, lines 42 – 51; col. 6, lines 17 – 27 of Kimura] stored in the second operating system interrupt table as to the point in the secondary operating system where the running of the interrupt is to take place [interrupt handler in the common area 203 captures the interrupt and decides an operating system to process the interrupt with reference to the processing OS 1621 of the interrupt identification table 1620. Then, the control is passed to the address of the handler 1622; col. 13, lines 46 – 56; col. 13, lines 24 – 25; col. 13, lines 45 – 63 of Kimura].

8. As to claim 32, Kimura as modified by Sekiguchi teaches a device for operating a secondary operating system [second OS; col. 5, lines 5 – 9 of Kimura] on a processor [col. 3, line 64 – col. 4, line 3 and col. 4, lines 20 – 22 of Kimura] in addition to a primary operating system [first OS; col. 4, lines 60 – 65 of Kimura] in which a change from the primary operating system to the secondary operating system takes place through an interrupt call [col. 16, lines 52 – 65; col. 13, lines 46 – 56; col. 15, line 58 – col. 16, line 5 of Kimura], the device comprising:

a function for replacing an interrupt table [interrupt table in the first OS main memory area; paragraphs 0152, 0146, and 0158 of Sekiguchi] of the primary operating system with an interrupt table [portions essential also for the second OS to access: such as the interrupt table 107; paragraphs 0218 and 0239 of Sekiguchi] of the secondary operating system [interrupt table in the first OS main memory area is changed to the interrupt table 107 of the support driver; paragraphs 0152, 0146, and 0158 of Sekiguchi];

a secondary operating system driver (SOS driver) of the primary operating system [control for running the multi-OS is incorporated as a device driver for the first OS (hereinafter referred to as multi-OS driver); col. 10, lines 19 – 22 and 54 – 60; col. 11, lines 14 – 20; col. 12, lines 4 – 10 of Kimura] for loading [loading the second OS by execution of the multi-OS management program in the first OS, Fig. 15; col. 12, lines 24 – col. 13, line 13 of Kimura] and controlling the secondary operating system [col. 12, lines 11 – 23 and 45 – 52; col. 14, lines 5 – 20; col. 15, lines 13 – 36 of Kimura], said secondary operating system driver comprising an interrupt call servicing routine [interrupt table 107, the interrupt management program, the interrupt handler, the interface module to be referred from each operating system and the like are stored in the common area as part of the multi-OS driver; col. 11, lines 21 – 32 of Kimura], wherein said interrupt call servicing routine determines the information [address of the interrupt handler of the second OS; col. 16, lines 42 – 51; col. 6, lines 17 – 27 of Kimura] stored in the interrupt table of the secondary operating system as to the point where interrupt running is to take place in said secondary operating system [interrupt

handler in the common area 203 captures the interrupt and decides an operating system to process the interrupt with reference to the processing OS 1621 of the interrupt identification table 1620. Then, the control is passed to the address of the handler 1622; col. 13, lines 46 – 56; col. 13, lines 24 – 25; col. 13, lines 45 – 63 of Kimura].

9. As to claim 2, Kimura teaches wherein the secondary operating system driver (SOS driver) subsequently loads the secondary operating system [col. 12, lines 24 – col. 13, line 23].

10. As to claim 3, Kimura teaches wherein the secondary operating system driver loads the secondary operating system [loading the second OS by execution of the multi-OS management program in the first OS, Fig. 15; col. 12, lines 24 – col. 13, line 13].

11. As to claim 4, Kimura teaches wherein memory contexts (virtual operating areas) are created in the central processing unit (CPU) [col. 4, lines 20 – 37].

12. As to claim 7, Kimura teaches wherein the secondary operating system controls a change to the primary operating system [col. 13, line 34 – col. 14, line 19].

13. As to claim 8, Kimura teaches wherein a change from the secondary operating system to the primary operating system takes place when the secondary operating

system is idle (entry into the idle loop) [first OS can run only when the second is idle; col. 13, lines 26 – 33]

14. As to claim 9, Kimura teaches wherein a change from the secondary operating system to the primary operating system takes place through an instruction in the program sequence of the secondary operating system [OS switching procedure; col. 14, lines 1 – 29].

15. As to claim 10, Kimura teaches wherein a change from the primary operating system to the secondary operating system takes place through an interrupt call [col. 16, lines 52 – 56].

16. As to claim 12, Kimura teaches wherein interrupt calls of the primary operating system are inhibited during the secondary operating system sequence [col. 15, lines 1 – 12].

17. As to claim 13, Kimura teaches wherein an interrupt servicing routine in the SOS operator reads the interrupt call table of the secondary operating system and the processing of the latter takes place or is continued at the point relative to the interrupt call [receives the address of the second OS module executed after switching from the first OS to the second OS; col. 14, lines 24 – 55].

18. As to claim 15, Kimura teaches wherein by means of an interrupt call servicing routine in the system driver, the information stored in the interrupt table of the secondary operating system (SOS) is determined as to the point in the latter where the running of the interrupt is to take place [col. 15, line 58 – col. 16, line 57 and col. 11, lines 21 – 32].

19. As to claim 17, Kimura teaches wherein after occurrence a corresponding interrupt call and determination of the point in the secondary operating system where interrupt running is to take place is determined, processing thereof at the point in the secondary operating system concerning the interrupt call is continued [col. 12, lines 54 – 67].

20. As to claim 18, Kimura teaches wherein on changing from one operating system to the other all the system states of one operating system are stored [OS context table; col. 13, lines 26 – 33].

21. As to claim 19, Kimura teaches wherein on changing from one operating system to the other all system states of the other operating system are loaded [col. 13, lines 34 – 45].

22. As to claim 21, Kimura teaches clock generation for the primary operating system takes place through a clock system driver [col. 4, lines 11 – 12].

23. **Claims 5, 11, 14, 16, 20, 23, and 25 – 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kimura and Sekiguchi further in view of U.S. Patent Application Publication No. 20040205755 to Lescouet et al. [hereinafter Lescouet].**

24. As to claim 5, Kimura teaches wherein a change between the operating systems takes place by means of the SOS driver of the primary operating system [col. 12, lines 24 – col. 13, line 13 and col. 14, lines 1 – 4]. Kimura and Sekiguchi do not teach a board support package (BSP).

However, Lescouet teaches switching between operating systems [paragraph 0088], a Board Support Package [paragraphs 0061 and 0064], and using the Board Support Package to change between operating systems [Board Support Package are specially adapted to assist the hardware resource dispatcher in virtualizing the shared devices for secondary operating systems; paragraph 0064].

It would have been obvious to a person of ordinary skill in the art to further modify the invention of Kimura and Sekiguchi to incorporate the features of Lescouet. One of ordinary skill in the art would have been motivated to make the combination because writing a new Board Support Package allow execution of the Operating System on a new computer with the same CPU but different system devices [paragraph 0061 of Lessouet].

25. As to claim 11, Kimura as modified teaches wherein the change between operating systems takes place by means of a program code filed in the tunnel area of the memory [paragraphs 0034, 0065, 0088 of Lessouet]. One of ordinary skill in the art would have been motivated to make the combination because this allows the primary operating system drivers to access shared resources, even if the access is requested by the secondary operating system [paragraph 0016 of Lessouet].

26. As to claim 14, Kimura as modified teaches wherein for each interrupt associated with the secondary operating system, which is to initiate an interrupt call in the secondary operating system, the system driver generates an entry in the interrupt call table in the primary operating system, which in turn initiates a call of the corresponding interrupt servicing routine in the secondary operating system [handling interrupts (using the real time operating system interrupt service routines, and supplying data where necessary to the virtual interrupt service routines of the secondary operating systems); paragraphs 0083, 0069, and 0141 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

27. As to claim 16, Kimura as modified teaches wherein in the case of activity of the secondary operating system (SOS) following an interrupt request through the information stored in the interrupt call table of the secondary operating system as to the point in the latter where the running of the interrupt is to take place [paragraphs 0083, 0069, and 0141 of Lessouet], the interrupt call servicing routine of the secondary

operating system (SOS) is directly polled solely by the secondary operating system and not via the system driver [paragraph 0138 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

28. As to claim 20, Kimura teaches wherein clock generation for the secondary operating system takes place through the hardware timer [clock interrupt generator 111; col. 4, lines 1 – 3 of Kimura and paragraph 0068 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

29. As to claim 23, Kimura as modified teaches wherein the SOS driver has a tunnel context setting routine for setting a tunnel context in the central processing unit (CPU) [paragraphs 0034, 0065, 0088 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

30. As to claim 25, Kimura as modified teaches wherein the SOS driver has an interrupt call table change routine for producing entries in the interrupt call table of the primary operating system, which at least take up entries for the interrupt calls for the secondary operating system [paragraphs 0083, 0069, and 0141 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

31. As to claim 26, Kimura as modified teaches wherein the board support package (BSP) has a section for return to the primary operating system (POS) [paragraphs 0061

and 0064 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

32. As to claim 27, Kimura as modified teaches wherein the secondary operating system driver (SOS driver) has an interrupt table [paragraphs 0083, 0069, and 0141 of Lessouet] section by means of which it produces in the primary operating system an interrupt call table containing a call of an interrupt servicing routine for polling the secondary operating system [paragraph 0138 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

33. As to claim 28, Kimura as modified teaches wherein the system driver is constructed for producing an entry in the interrupt call table in the primary operating system (POS) for each interrupt associated with the secondary operating system (SOS), which is intended to initiate an interrupt call in the secondary operating system and that the interrupt call table is constructed for polling the corresponding interrupt servicing routine in the secondary operating system (SOS) [paragraphs 0083, 0069, and 0141 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

34. As to claim 29, Kimura as modified teaches wherein an interrupt call servicing routine in the system driver is constructed for determining the information stored in the SOS interrupt table as to the point in the secondary operating system where interrupt

running is to take place [paragraphs 0083, 0069, and 0141 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

35. As to claim 30, Kimura as modified teaches wherein it is constructed in the case of activity of the secondary operating system (SOS) following an interrupt request through the information stored in the secondary operating system interrupt call table as to the point in which the secondary operating system interrupt running is to take place, so as to poll the interrupt call servicing routine of the secondary operating system directly solely through the secondary operating system and without passing via the system driver [paragraphs 0083, 0069, and 0141 of Lessouet]. As to the reasons for combining Kimura and Lessouet, see the rejections for claims 5 and 11.

### ***Conclusion***

36. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

**CONTACT INFORMATION**

37. Any inquiry concerning this communication or earlier communications from the examiner should be directed to LI B. ZHEN whose telephone number is (571)272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sub Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/
Primary Examiner, Art Unit 2194